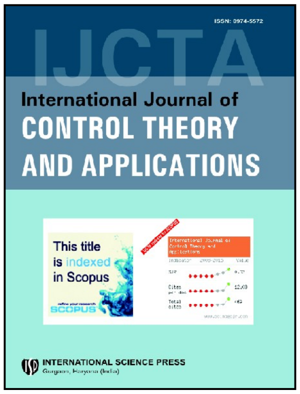

International Journal of Control Theory and Applications

ISSN : 0974-5572

Volume 10 • Number 10 • 2017

Predicting the Stock Price Index of Yahoo Data Using Elman Network

S. Kumar Chandar

Associate Professor, Christ University, Bangalore, India, E-mail: kumar.chandar@christuniversity.in

Abstract: Accurate stock market prediction is the challenging area for the businessmen to yield profits in the financial markets. The investors need to understand the financial markets which are more volatile and affected by many external factors. Hence, an Artificial Neural network is one of the powerful tools for predicting the stock prices. This paper proposes ELMAN back propagation networks for predicting the Yahoo stock data. The period used for study is from 1st January 2009 to 31st April 2014. Ten technical indicators are proposed in this study. They are Simple moving average for 1 week, Simple moving average for 2 weeks, Momentum, Rate of change, 7 day Disparity, 14 day Disparity , Price Oscillator, Accumulation or distribution Oscillator, Larry Williams R%, Commodity channel index. These variables are given as inputs for predicting the daily closing values of yahoo stock prices. Also, this study compares the three models namely Feed forward networks without feedback, Feed forward back propagation networks and ELMAN networks. The accuracy measures like Mean Square Error, Mean Absolute Error, Sum Square Error and Root Mean Square Error are used to access the performance. From the results, it is clear that ELMAN network outperforms the other two models.

Keywords: Yahoo stock data, ELMAN back propagation network, Technical Indicators, Financial Markets, Artificial Neural Network

1. INTRODUCTION

In the recent years, Stock market prediction is one of the vital issues in the financial analysis. The investors want to know the future market in order to avoid risk that occurs at any time due to chaotic behaviour of the market. Also, they desire to achieve profit from their money. Hence, the future stock market prediction becomes the caution system for both short-term and long-term investors against the unexpected threat in the market. Thus, many researches are carried on the accurate stock market prediction of the financial data.

Computational Intelligence techniques are emerged to tackle the difficulties in the prediction of stock markets. Some of the techniques are Artificial Neural Networks (ANN), Support Vector Machines, Genetic Algorithms, Fuzzy logic and so on. Artificial Neural networks suits very well for the stock market prediction. ANN techniques are very robust when compared with the other techniques, which makes the prediction more accurate. A trained neural network can detect future trends that are too complex to be detected by Humans or other intelligence techniques.

Literature studies on applications of various techniques for prediction of stock markets are discussed. Lijuan Cao *et al.* [1] used SVM model for the prediction of S&P 500 daily price index. They proved that SVM model is better in terms of Normalized mean square error and mean absolute error. Kyoung-jae Kim *et al.* [2] proposed Support Vector Machine for financial time series prediction. Several initial features are selected as inputs to the networks. Also, they compared SVM models with the ANN model and Case based reasoning model. Birol Yildiz *et al.* [3] proposed ANN models to forecast the direction of Istanbul Stock Exchange. Their results showed that ANN models produce an accuracy of 74.51%. Yakup Kara *et al.* [4] used artificial neural networks and Support Vector Machine for the prediction of Istanbul Stock Exchange. Their models used ten technical indicators as inputs to their networks. Their results showed that the ANN models were remarkably improved than the SVM model.

Zabir Haider Khan *et al.* [5] used Back propagation algorithm for training and Multilayer Feed forward network as a network model for price prediction of share market. Adebiyi Ayodele *et al.* [6] used hybridized approach for predicting the stock prices. They proved that the hybridized approach was considerably better than the other techniques. Neelima Budhani *et al.* [7] used Feedforward backpropagation network for the prediction of stock market. Their training procedure makes the better improvement in the prediction of stock markets Victor Devadoss *et al.* [8] suggested ANN models for prediction of closing values of Bombay stock exchange. The inputs to the networks are high, low, opening price, closing price, and volume. Root mean square error, Mean Absolute Deviation and Mean Absolute Percentage Error are used as performance indicators for the network. Yanshan Wang et al [9] proposed machine learning techniques like Principal Component Analysis (PCA) for identifying the principal components and Support Vector Machines used for classifier for future stock market movement. They considered Korean composite stock price index (KOSPI) and Hangseng Index (HSI) as stock data for their study. Najeb Masoud et al [10] used ANN model for predicting the direction of Libyan Financial market. Mean square error and Root mean square errors are used as performance indicators.

Nitin Anand Shrivastav *et al.* [11] proposed ELM model for predicting the accurate electricity markets forecasting. Their results showed that their model was one of the most suitable predicting techniques. Lucas Lai *et al.* [12] used Support Vector Machine and Least Square Support Vector Machine for forecasting the stock prices. Qisheng Yan *et al.* [13] proposed Extreme Learning Machine with Empirical mode decomposition for forecasting Uranium resource price.

The organization of the paper is as follows: Section 1 briefs on the Introduction and Literature survey. Section 2 explains the research data used in this study; Section 3 explains the application of ANN techniques like Feed forward networks without feedback, Feed forward back propagation networks and ELMAN network to yahoo stock data. Section 4 briefs on the discussion and results, Section 5 concludes the paper and Section 6 explains on the reference.

2. RESEARCH DATA AND TECHNICAL INDICATORS

2.1. Research data

The section explains the research data used in this study. This yahoo stock data is collected from *www.finance.yahoo.com.* This work proposes to predict the daily change in the Yahoo stock market and finds the future predicted value. The period of the study is from 1st January 2004 to 31st April 2014. The entire data is divided into training dataset, validation dataset and testing dataset. Totally 1340 data is retrieved and among that, training dataset have 938 data (70%) and validation dataset have 201 data (15%) and testing dataset have 201 data (15%). The sample data is shown below in table 1.

Table 1
A sample of Yahoo stock market data

Date	*Open*	*High*	*Low*	*Close*	*Volume*	*Adj Close*
02-01-2009	12.17	12.85	12.12	12.85	9514600	12.85
05-01-2009	12.72	13.01	12.39	12.86	11989900	12.86
06-01-2009	12.96	13.24	12.88	13	10056000	13
07-01-2009	12.71	13.16	12.45	12.71	24995900	12.71
08-01-2009	12.37	13.07	12.31	13.07	14355000	13.07
09-01-2009	13.42	13.56	12.9	13.13	19281000	13.13
12-01-2009	13.09	13.1	12.08	12.22	19976900	12.22
13-01-2009	12.09	12.79	11.78	12.1	25720400	12.1
14-01-2009	12.26	12.53	11.81	12.41	23595200	12.41
15-01-2009	12.32	12.35	11.22	11.61	25247500	11.61
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.

This sample data have attributes like date of stock data, stock opening price, highest price in that date, lowest price in that date, stock closing price, volume and adjacent close. This study uses stock opening price, stock highest price, stock lowest price and stock closing price.

2.2. Technical Indicators

The investors use technical indicators for predicting the stock prices. Various technical indicators are available. In this research study, ten technical indicators are selected from the literature papers (Yakup Kara et al. [4] and Kyoung-jae Kim [2]). Using the formula listed in table 2, the input attributes are calculated and given as input to the neural networks which in turn gives us the predicted next day's stock price. The technical indicators used in this research are tabulated below in table 2.

Table 2
Technical indicators and its formula

S.No	*Technical Indicators*	*Description*	*Mathematical formula*	*Ref*
1	Simple moving average for 1 week	Average of currency rates for 1 week	$\frac{\mathbf{C_t + C_{t-1} + C_{t-2} + C_{t-3} + C_{t-4} + C_{t-5} + C_{t-6} + C_{t-7}}}{\mathbf{7}}$	[4]
2	Simple moving average for 2 week	Average of currency rates for 2 weeks	$\frac{\mathbf{C_t + C_{t-1} + C_{t-2} + \ldots\ldots + C_{t-14}}}{\mathbf{14}}$	[4]
3	Momentum	Gives the quantity that the currency rates have altered over a time t	$C_t - C_{t-4}$	[4]
4	Price rate of change (ROC)	Gives the difference between the current rate and the rate at 4 days ago.	$\frac{\mathbf{C_t * 100}}{\mathbf{C_{t-4}}}$	[2]
5	Disparity 7(7 day disparity)	It displays the distance of current rate and the moving average of 7 days	$\frac{\mathbf{C_t * 100}}{\mathbf{MA\ 7}}$	[2]

contd. table 2

S.No	Technical Indicators	Description	Mathematical formula	Ref
6	Disparity 14(14 day disparity)	It displays the distance between the current rate and the moving average of 14 days	$\frac{\mathbf{C_t * 100}}{\mathbf{MA\ 14}}$	[2]
7	Price Oscillator	Distance between moving average of 7 days and moving average of 14 days	$\frac{\mathbf{MA\ 7 - Ma\ 14}}{\mathbf{MA\ 7}}$	[2]
8	Accumulation / Distribution Oscillator	It links the changes in price	$\frac{\mathbf{H_t - C_{t-1}}}{\mathbf{H_t - L_t}}$	[2]
9	William's R%	Momentum indicator	$\frac{\mathbf{H_n - C_t}}{\mathbf{H_n - L_n}} * 100$	[2]
10	CCI	Commodity Channel Index.	$\frac{\mathbf{S_t - SS_t}}{\mathbf{0.015K_t}}$ $S_t = \frac{C_t + L_t + C_t}{3}$ $SS_t = \sum_{i=1}^{n} \frac{S_{t-i+1}}{n}$ $\mathbf{K_t} = \sum_{i=1}^{n} \frac{S_{t-i+1} - SS_t}{n}$	[2]

H_t is the highest price at time t, L_t is the lowest price at time t, C_t is the closing price at time t, MA is the moving average

The summary statistics for the research data is calculated from the technical indicators by minimum value, maximum value, mean and standard deviation which are given in table 3.

Table 3
Summary statistics of the research data from technical indicators

Technical Indicator	*Max*	*Min*	*Mean*	*Standard Deviation*
Simple moving average for 1 week	40.89857	1.835714	19.06884	7.210919
Simple moving average for 2 week	40.55071	0.917857	18.97623	7.206371
Momentum	13	-3.57	0.078933	0.926576
Price rate of change (ROC)	120.8808	0	100.065	5.939378
Disparity 7(7 day disparity)	700	87.97452	101.0723	18.47157
Disparity 14(14 day disparity)	1400	85.0367	102.8639	41.67209
Price Oscillator	0.5	-0.08732	0.005999	0.044827
Accumulation / Distribution Oscillator	17.60274	-1.94118	0.548499	0.703424
William's R%	100	0	50.57009	29.16897
CCI	133.3333	-133.333	5.768234	88.2103

3. PREDICTION MODELS

An artificial neural network is an information processing system that resembles the working of human nervous system. It is composed of a large number of interconnected processing elements called neurons which are operated in parallel to solve specific problems [14]. Each neuron is connected with other neurons by a connection link called weights which contain information about the input signal. The bias is added to provide balance to the network. The input signals are multiplied with their corresponding weights. These products are summed to get the net input. The output of the network is obtained by applying activations over the net input. Several activation functions are available which are Identity function, binary step function, bipolar step function, sigmoidal functions and so on.

The main advantage of neural network is the capability to learn. There are three types of learning. They are supervised learning, unsupervised learning and reinforcement learning. Supervised learning works with the help of a teacher. Here both the training pair and the targets are given to the network and the network will learn according to the target pair. Some of the examples of supervised learning are Perceptron networks, Adaptive linear neuron networks, multiple adaptive linear networks, Back propagation networks, Radial basis function networks and so on. In unsupervised learning, the learning process is not supervised by a teacher. In this type of learning, when a new input pair is given to the network, the network gives an output belonging to a class. The examples of unsupervised learning are Kohonen self-organising feature maps, Learning Vector Quantization and Adaptive resonance theory networks. In reinforcement learning, the process is similar to supervised learning. Only a part of the information is known to the network. In this work supervised learning networks are considered. The proposed architecture is shown in figure 1. In this figure, the inputs for the neural networks are the attributes values of technical indicators. They are Simple moving average for 1 week, Simple moving average for 2 weeks, Momentum, Rate of change, 7 day Disparity, 14 day Disparity , Price Oscillator, Accumulation or distribution Oscillator, Larry Williams R%, Commodity channel index. These variables are given as inputs for predicting the next day's closing values of yahoo stock prices.

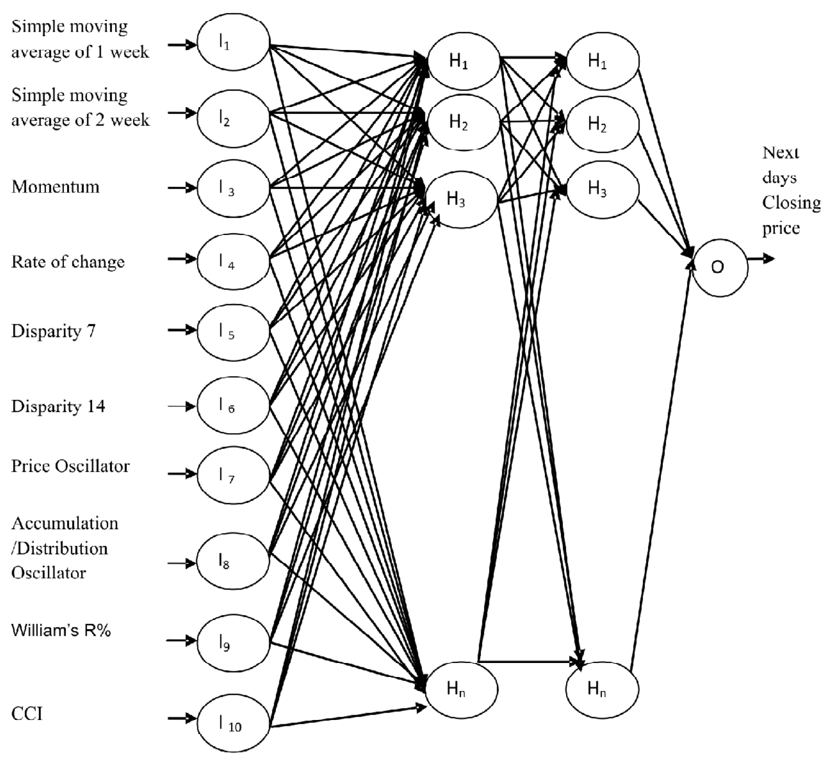

Figure 1: Architecture of the proposed system

This section explains the various prediction models like Feed forward networks without feedback, Feed forward back propagation networks and ELMAN networks.

3.1. Feed forward networks without feedback

A network is said to be feed forward network when the outputs are not directed back as inputs to the same or preceding layer. In this network, the information moves only in one direction and there is no feedback. This model tends to find the next day's yahoo stock price. This subsection explains the architecture, training and regression analysis of the Feed forward networks.

3.1.1. Architecture

The architecture of feed forward network is shown in Fig.2. The network is two layers feed forward network consisting of an input layer, two hidden layers and one output layer. Initially, ten technical indicators as discussed above are given as inputs to the first hidden layer of feed forward networks. When the input signals are presented to the network, they are multiplied with their corresponding weights to produce the net input. Bias is added to both hidden layers to improve the performance of the network. The first hidden layer calculates the output by applying activation function to the net input of the first hidden layer. Then the output of the first hidden layer is given as inputs to the second hidden layer. The output of the network is obtained by applying activation function over the calculated net input of second hidden layer. There are several transfer functions like TANSIG, LOGSIG, or PURELIN. But in this model, TANSIG function is used. All the layers are trained using Levenberg-Marquardt (TRAINLM) function and INITNW and TRAINS functions are used for adaptation of weights.

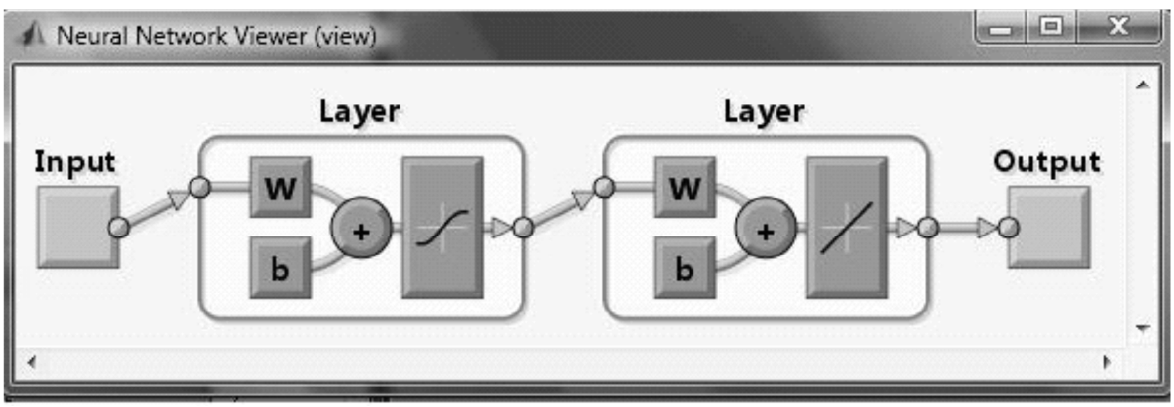

Figure 2: Feed forward networks without feedback

3.1.2. Training the Feed forward networks without feedback

The network training tool for feed forward networks without feedback is shown in Fig.3. This tool has four sections namely the structure of the network, algorithms used, the training progress of the network and the various plots. During training process, this network has been given with 1000 epochs. But, it achieves the best performance of 0.149 at 14th iterations.

In the plots section of above figure, the four plots can be viewed. When Performance button is pressed, performance graph can be viewed. Similarly, other plots can also be viewed. The performance graph for this model is shown in Fig. 4. This graph shows training, validation and testing errors. This curve plots between the number of epochs and mean square error. The mean square error is the error obtained at each iteration. This error should be minimized in order to get the best performance. The best validation performance is achieved at 0.37436 at 8th iteration.

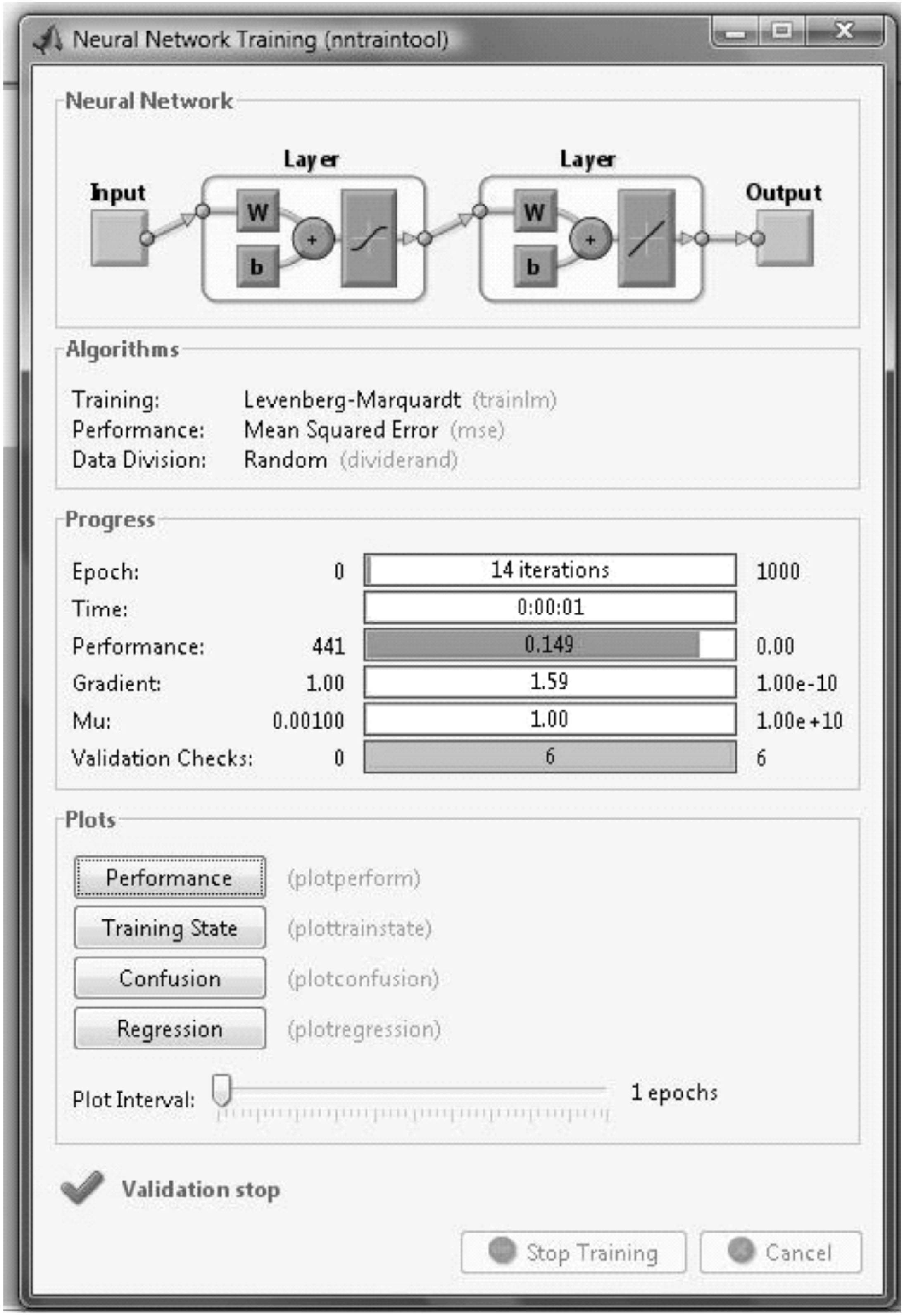

Figure 3: Training process of Feed forward networks without feedback

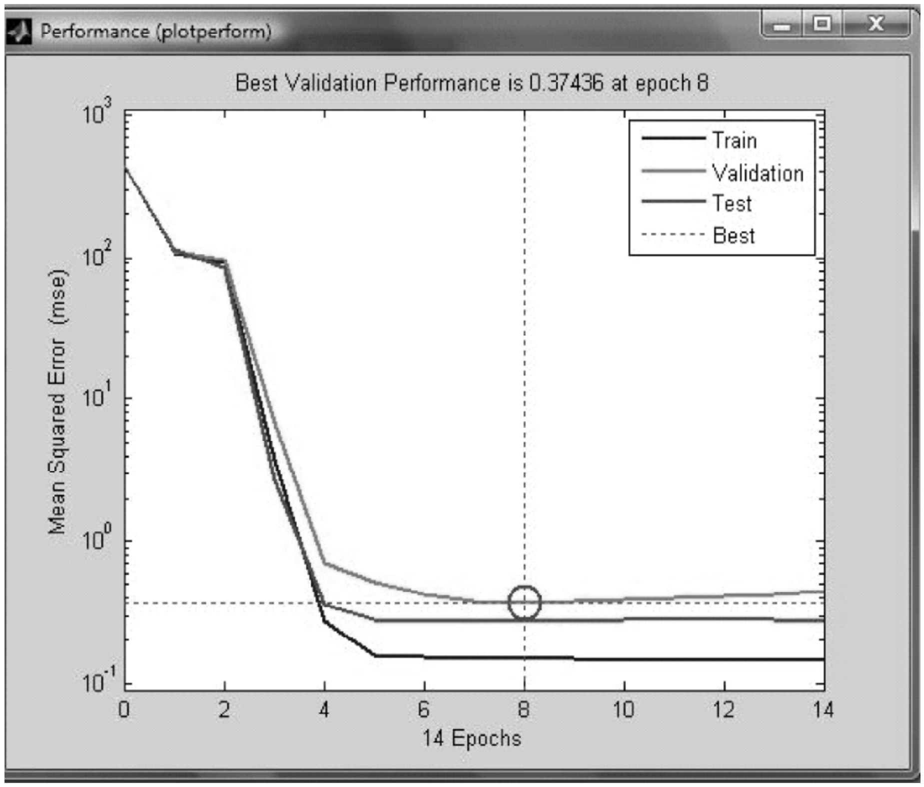

Figure 4: Performance curve of Feed forward networks without feedback

The confusion matrix for this model is shown in Fig. 5. This figure shows four matrixes. They are training, validation, testing and all confusion matrixes. In training process, among 938 data used for training, only 469 data has been classified correctly. Also in the validation process, only 50.2 % of the data has been positively classified among 201 data. And in the testing process, the same 50.2% have been classified correctly among 201 data. As a whole, only 50.0% have been classified correctly and the remaining 50.0% have been wrongly classified.

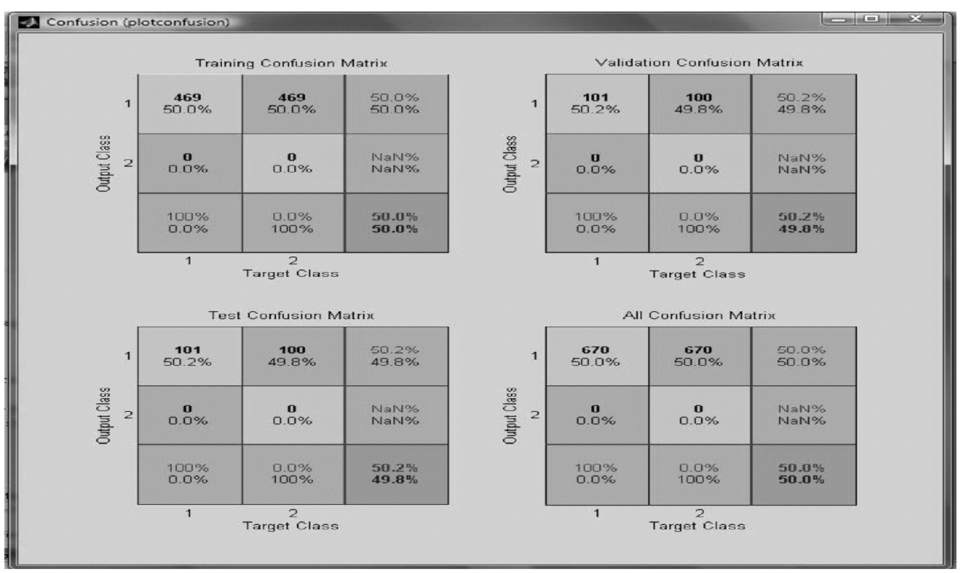

Figure 5: Confusion matrix of Feed forward networks without feedback

3.1.3. Regression analysis of Feed forward networks without feedback

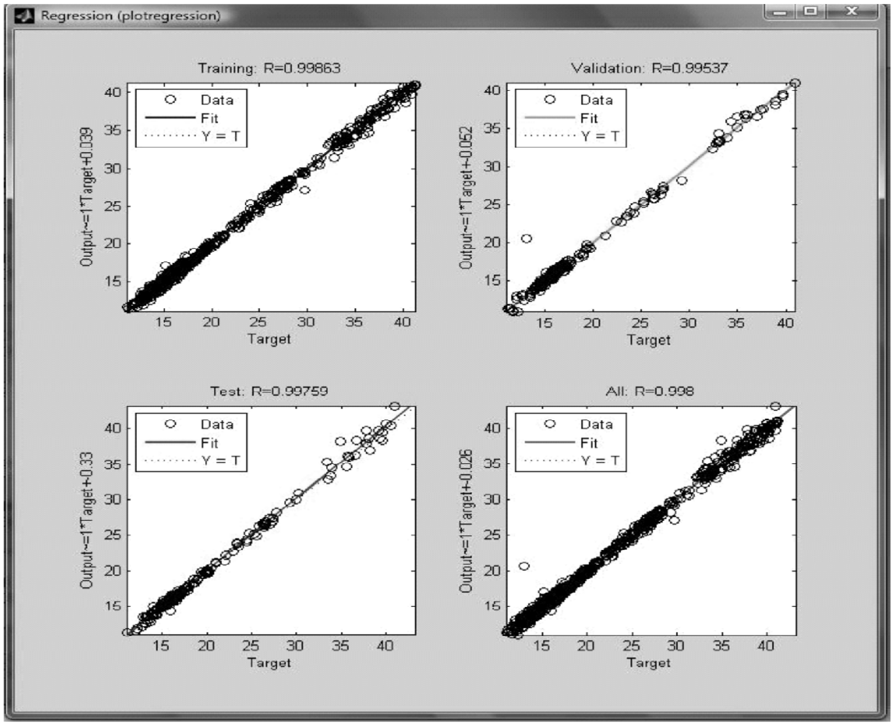

Figure 6: Regression analysis of Feed forward networks without feedback

The regression is the process of fitting the models with the data. The regression analysis is shown in Fig 6. During training process, this model has achieved 0.99863 R-values and in the validation process, it has achieved 0.99537 R-value and the testing process has achieved 0.99759. As a whole, this model has achieved 0.998 R-values which is closer to 1. This shows that this model has fitted more closely to the target data.

3.2. Feed forward back propagation networks

Back propagation networks are multi-layered feed forward networks trained with respect to the back propagation error algorithm. This is the most widely used network. Here, the outputs are directed back as inputs to the same or preceding layer nodes. This subsection discusses the architecture, training process and regression analysis of back propagation networks.

3.2.1. Architecture

The architecture of feed forward back propagation network is shown in Fig 7. This model also uses 1 input layer, two hidden layers and one output layer.

The training process of back propagation networks is divided into three stages. They are

1. Feed forward of input patterns
2. Calculation of Back propagation error
3. Updating the weights

This network is a two-layer feed-forward network with a two-delay input and two-delay feedback is created. During Feed forward process, the ten technical indicators are given as inputs to the input layer. The input layer then feeds these signals to first hidden layer. The outputs obtained from the first hidden layer are sent to second hidden layer. The back propagation errors are calculated and these errors are sent in reverse direction. On the basis of calculated errors, the weights are updated.

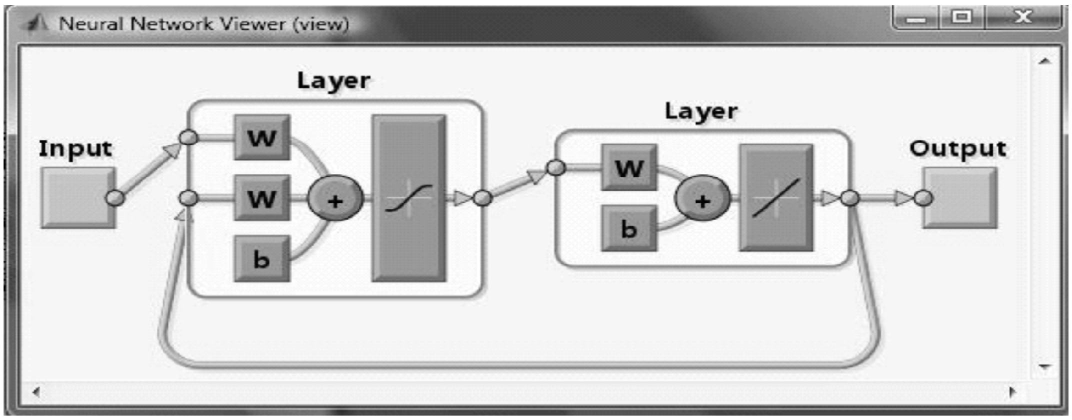

Figure 7: Architecture of Feed forward back propagation networks

3.2.2. Training of Feed forward back propagation networks

The training tool of Back propagation network is shown in Fig, 8. The input layer has 10 neurons and both the hidden layers have 5 neurons and output layer has 1 neuron to produce the output. Normalization process is required for the dataset to remove noise and outliers. Hence in this model, the function '**MAPMINMAX**' is used to reduce the noise in the dataset. Training is done with the Levenberg-Marquardt '**TRAINLM**' training function and back propagation weight / bias learning is done using '**LEARNGDM**' function. Adaption is done with '**TRAINS**' which updates weights with the specified learning function. The transfer function '**TANSIG**' function is used for hidden layers and '**PURELIN**' function for output layer. Performance is measured according to the mean square error '**MSE**' performance function. This model runs up to 39 iterations and achieves the best performance goal.

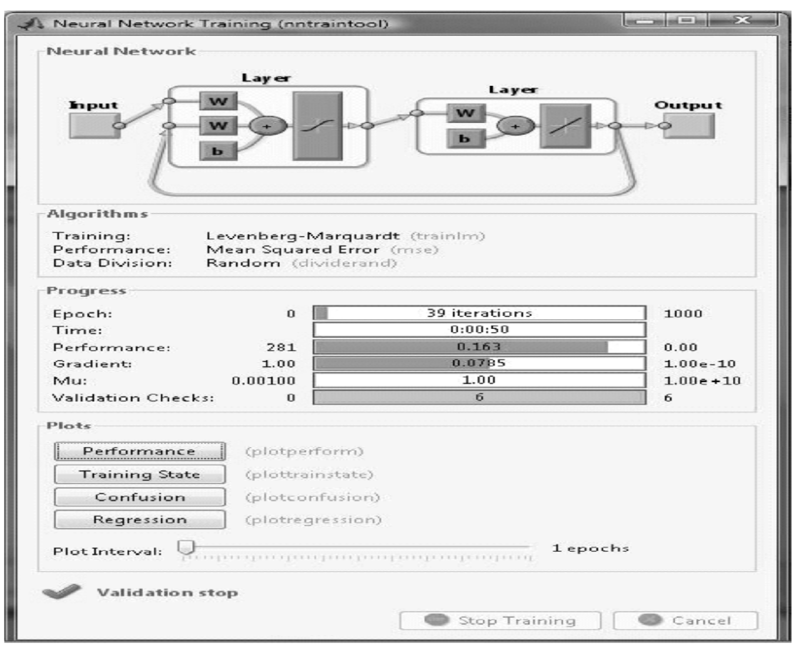

Figure 8: Training of Feed forward back propagation networks

The performance curve for the feed forward back propagation network is depicted in Fig. 9. This curve shows that best validation performance is achieved of value 0.27504 at 33rd iteration.

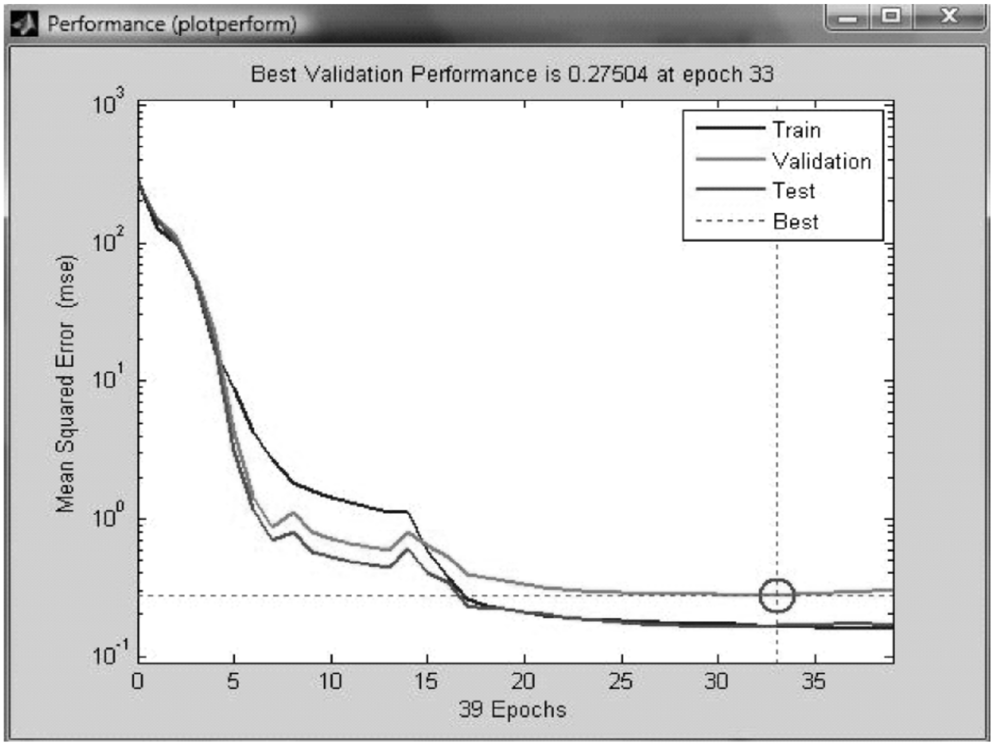

Figure 9: Performance curve of Feed forward back propagation networks

3.2.3. Regression Analysis of Feed forward back propagation networks

The regression plots for feed forward back propagation networks are shown in Fig.10. This plot shows that training process achieves 0.99844 R-values and validation process achieved 0.99752 and testing process achieves 0.99843. Overall 0.99828 R-value is achieved for this model which is closer to 1.

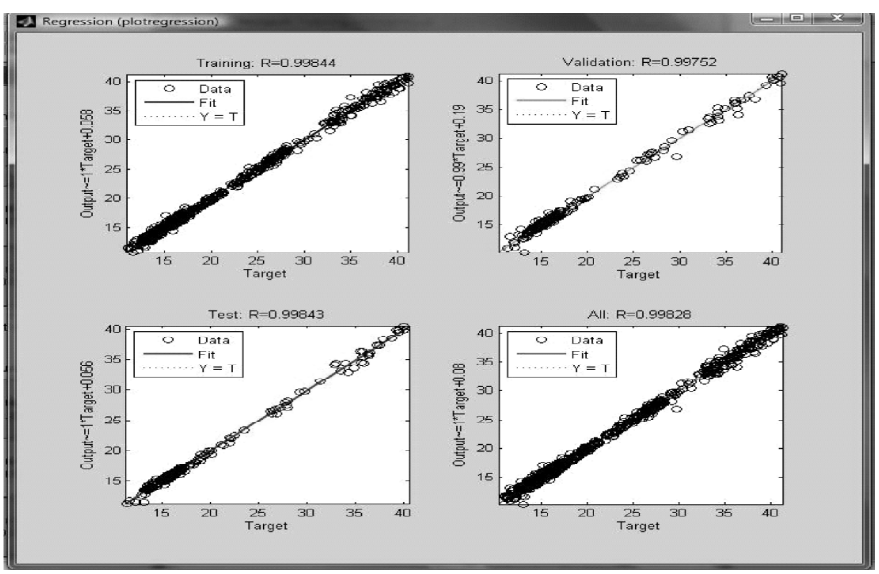

Figure 10: Regression analysis of Feed forward back propagation networks

3.3. ELMAN Networks

3.3.1. Basics

The ELMAN networks are a form of recurrent neural networks developed by Jeffrey L. Elman. A three layer feed forward network where a set of 'Context Units' are attached to the hidden layer. At each time step, a copy of hidden layer unit is copied onto the Context unit having a weight of one. Thus the network is learnt by the current input signals, a copy of previous hidden layer unit's i.e context units and the output of the network. The context unit can be considered as a one of the inputs to the hidden layer. Jordan networks are similar to ELMAN networks where the context units are fed from the output layers instead from the hidden layer. Both the networks are referred to as Simple Recurrent networks. This subsection explains the architecture, algorithm, training and performance analysis of the ELMAN networks.

3.3.2. Architecture

The architecture of ELMAN network is shown in Fig 11. The ten technical indicators are given as inputs to the input layer. This layer contains 10 neurons. These signals are fed into the first hidden layer where the signals from context units are also passed in to this layer. Here 10 hidden neurons are used. The hidden layer activations are done and a copy of hidden layer units is saved into the context units. The outputs of the first hidden layer are given to the second hidden layer where in turn produces the output which is given to output layer.

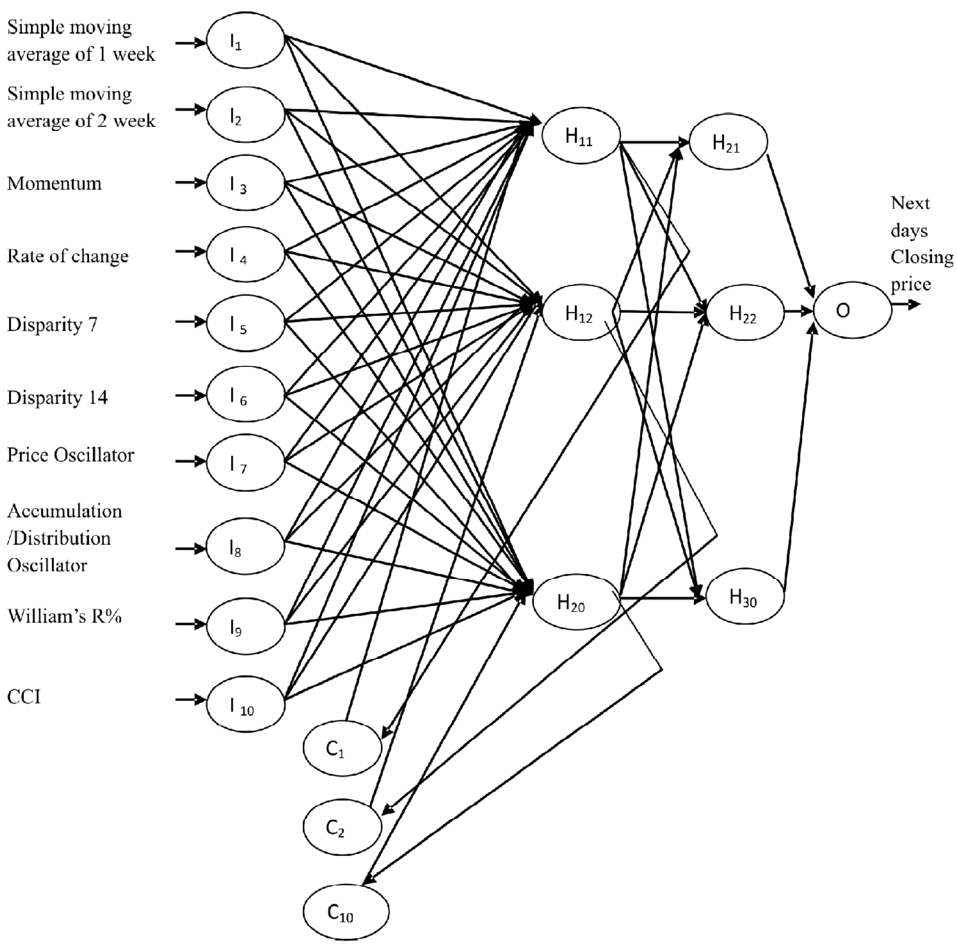

Figure 11: Architecture of proposed ELMAN network

3.3.3. Algorithm

The algorithm of ELMAN networks is given below:

Step 1: First initialize the weights. The activations of the context units are set to 1.

Step 2: Perform steps 2- 9 until when stopping condition is false

Step 3: Perform steps 3-10 for each training pair

Step 4: Get the input signals from the input layer and present these signals to the hidden units.

Step 5: Each hidden units sums its weighted input signals to calculate net input of the hidden layer.

Step 6: Calculate the output of the hidden unit by applying activation function over the net input. Now, the output signal from the hidden layer is given to output layer.

Step 7: Each output unit calculates the net input by summing the weighted input signals of the hidden layer. Now the output of the output unit is found by applying activations over it.

Step 8: Each output unit finds the back propagation error.

Step 9: Check if the target pattern matches with the output of the output layer. If it matches stop the process, else copy the hidden unit activations to the context units and continue from step 3.

Step 10: The stopping condition may be the number of epochs or when the target matches with the actual output.

3.3.4. Training the ELMAN networks

There are several training function like **TRAINGD** (Training with Gradient Descent), **TRAINGDM** (Training with Gradient Descent with momentum), **TRAINGDA** (Training with Gradient Descent with adaptive learning rate), and **TRAINGDX** (training with gradient descent momentum and an adaptive learning rate) whereas in this paper, **TRAINGDX** function is used. Similarly, there are several learning functions like **LEARNGD** (Learning with Gradient descent weight and bias) or **LEARNGDM** (Learning with Gradient descent with momentum weight and bias) function. For this model, **LEARNGDM** function is used. The performance function used in this model is **MSE** (Mean Square Error). Each layer's weights and biases are initialized with **INITNW**. Adaption is done with **TRAINS** which updates weights with the specified learning function. The transfer function **TANSIG** function is used for hidden layers and **PURELIN** function for output layer. The training of ELMAN network is shown in Fig. 12. This network runs upto 1000 iterations and achieves best performance goal.

The performance curve for the ELMAN network is depicted in Fig. 13. This curve shows that best validation performance is achieved of value 0.292 at 994^{th} iteration.

3.3.5. Regression analysis of ELMAN network

The regression plots for ELMAN networks are shown in Fig.14. This plot shows that this network achieves 0.99721 R-values which is closer to 1.

4. PERFORMANCE COMPARISON

The following performance indicators are used to predict the output response.

1. Mean Square Error (MSE)
2. Mean Absolute Error (MAE)
3. Sum Squared Error (SSE)
4. Root Mean Square Error (RMSE)

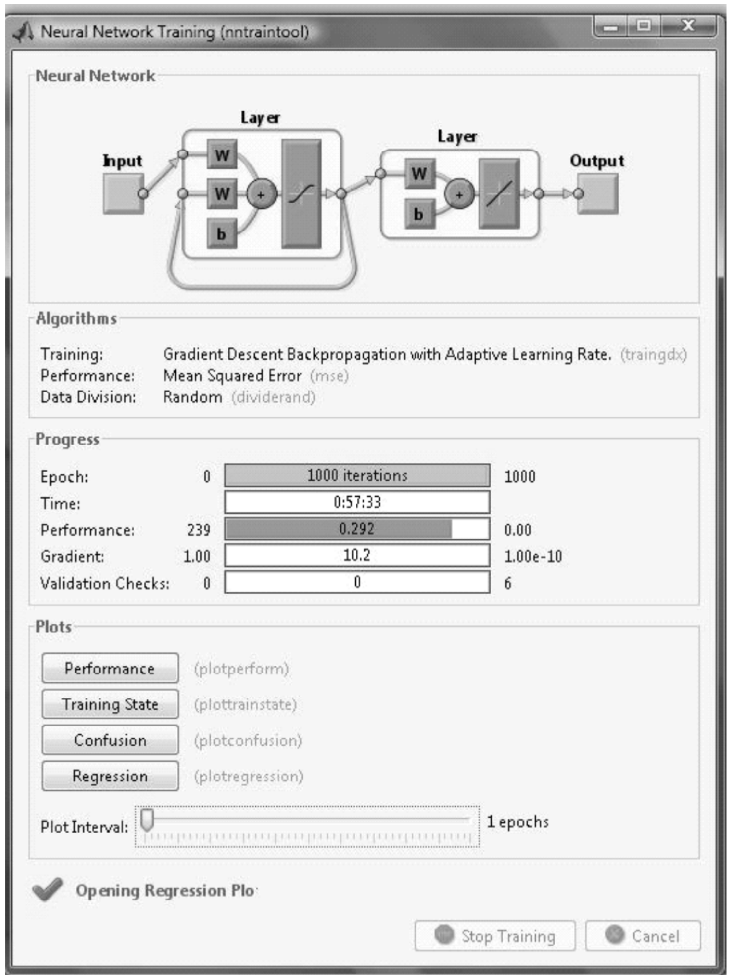

Figure 12: Training the ELMAN network

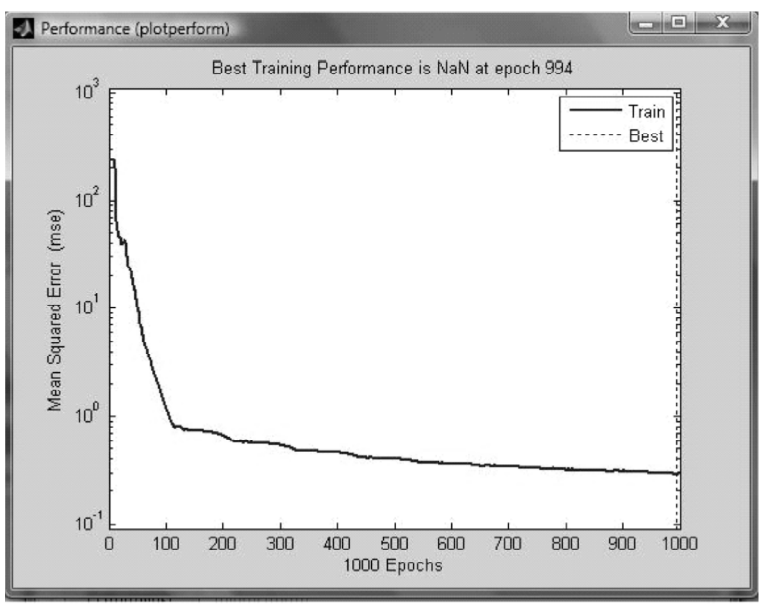

Figure 13: Performance curve of Feed forward back propagation networks

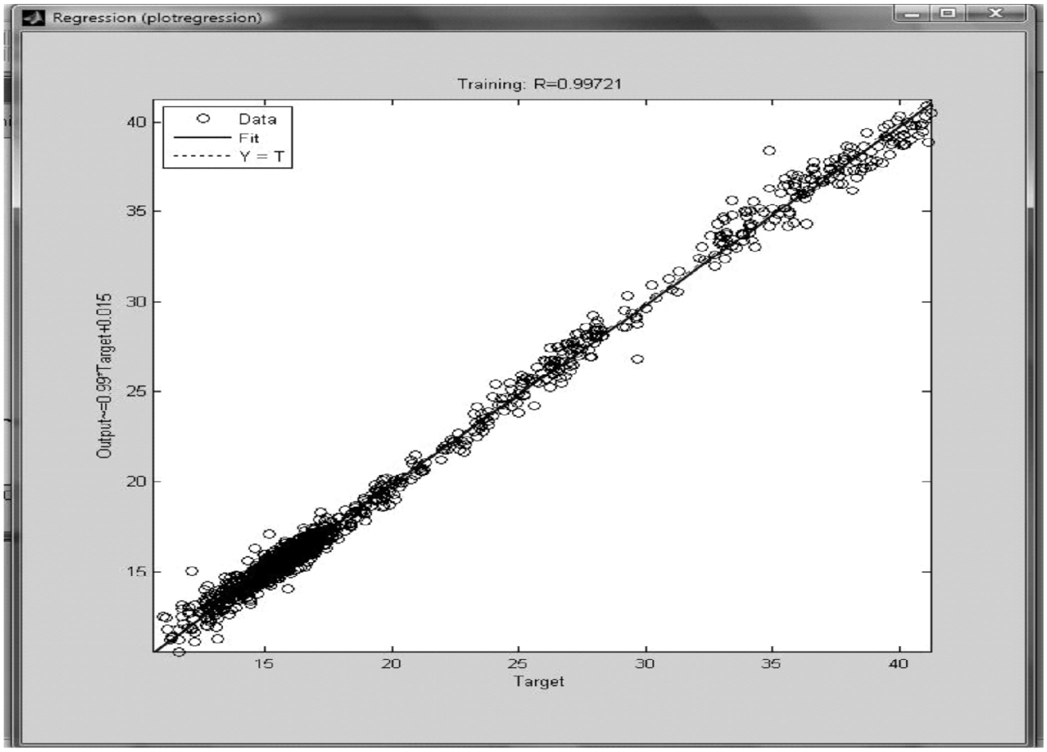

Figure 14: Performance curve of ELMAN networks

The proposed model is compared with respect to Mean square error, Mean absolute error, Sum squared error and Root mean square error. This model is also compared with the literature paper Kyoung-jae Kim et al [2]. The performance comparison of these models is shown in table 4.

Table 4
Performance comparison of proposed model with the existing models

	Feed forward networks	*Feed forward networks with feedback*	*ELMAN networks*	*Kyoung-jae Kim[2]*
Mean Square Error	0.2020	0.1800	0.1021	0.1098
Mean Absolute Error	0.2874	0.2923	0.1895	0.2375
Sum Square Error	270.6751	241.1958	154.8032	207.8453
Root mean Square error	0.4494	0.4243	0.3196	0.3045

From the table, it is inferred that the ELMAN network has 0.1021 mean square error value which is smaller than feed forward networks, feed forward networks with feedback and Kim model. Also, it achieves 0.1895 of mean absolute error value, 154.8032 of sum square error value and 0.3196 of root mean square error value. All the calculated error values are smaller than the other models. Hence, the proposed model outperforms the other existing models. The graph is drawn between yahoo stock data and time series which is shown in Fig.15. In this graph, the predicted yahoo stock prices are mapped and it is compared with the actual stock prices. From the graph, it is understood that this proposed model predicts the future stock prices more accurately with the actual stock prices.

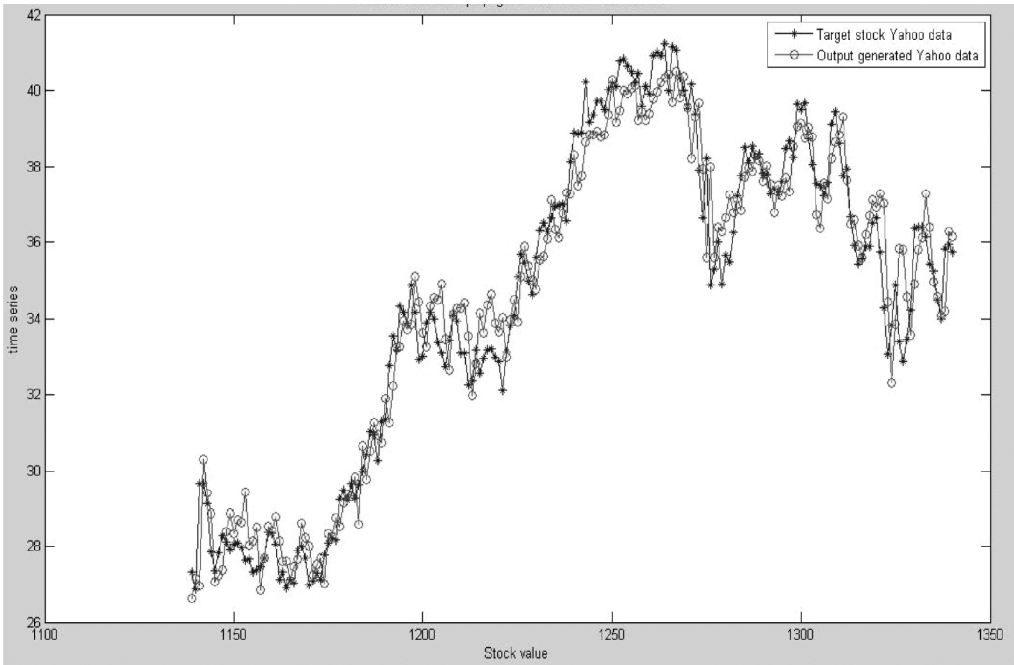

Figure 15: Comparison graph of Predicted stock Vs Target stock price

5. CONCLUSION

This paper proposes the ELMAN network for predicting the next day's stock price of Yahoo data. Also, this paper proposes ten technical indicators like Simple moving average for 1 week, Simple moving average for 2 weeks, Momentum, Rate of change, 7 day Disparity, 14 day Disparity , Price Oscillator, Accumulation or distribution Oscillator, Larry Williams R%, Commodity channel index as inputs to the network. Also, this study compares the three models namely Feed forward networks, Feed forward back propagation networks and ELMAN networks. The accuracy measures like Mean Square Error, Mean Absolute Error, Sum Square Error and Root Mean Square Error are used to access the performance. From the results, it is clear that ELMAN network outperforms the other two models.

REFERENCES

[1] Lijuan Cao and Francis E.H. Tay, Financial Forecasting Using Support Vector Machines, Neural Computing & Applications, Vol 10, 2001, pp. 184–192.

[2] Kyoung-jae Kim, Financial time series forecasting using support vector machines in the Neurocomputing, Vol. 55, 2003, pp. 307 – 319.

[3] Birol Yildiz, Abdullah Yalama, and Metin Coskun, Forecasting the Istanbul Stock Exchange National Index Using an Artificial Neural Network in the World Academy of Science, Engineering and Technology, Vol 2, 2008, pp.10-23,

[4] Yakup Kara, Melek Acar Boyacioglu, Ömer Kaan Baykan, Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange in the Expert Systems with Applications, Vol 38, 2011 pp. 5311–5319.

[5] Zabir Haider Khan, Tasnim Sharmin Alin, Md. Akter Hussain, Price Prediction of Share Market using Artificial Neural Network (ANN) in the International Journal of Computer Applications, Vol 22, No.2, May 2011, pp. 42-47.

[6] Adebiyi Ayodele A., Ayo Charles K., Adebiyi Marion O., and Otokiti Sunday O, Stock Price Prediction using Neural Network with Hybridized Market Indicators in the Journal of Emerging Trends in Computing and Information Sciences, Vol.3, No.1, 2012, pp. 1-9.

[7] Neelima Budhani, C. K. Jha, Sandeep K. Budhani, Application of Neural Network in Analysis of Stock Market Prediction in the International Journal of Computer Science & Engineering Technology, Vol. 3 No. 4 April 2012,pp 61-68.

[8] Victor Devadoss, T. Antony Alphonnse Ligori, Stock Prediction Using Artificial Neural Networks in the International Journal of Data Mining Techniques and Applications, Vol 2, 2013, pp.283-291.

[9] YanshanWang, In-Chan Choi, Market Index and Stock Price Direction Prediction using Machine Learning Techniques: An empirical study on the KOSPI and HSI in the Science Direct, Vol. 00, 2013, pp.1–13.

[10] Najeb Masoud, Predicting Direction of Stock Prices Index Movement Using Artificial Neural Networks: The Case of Libyan Financial Market in the British Journal of Economics, Management & Trade, Vol. 4, No.4, 2014, pp.597-619.

[11] Nitin Anand Shrivastava, Bijaya Ketan Panigrahi, A hybrid wavelet-ELM based short term price forecasting for electricity markets in the Electrical Power and Energy Systems, Vol. 55, 2014, pp. 41–50.

[12] Lucas Lai, James Liu, Support Vector Machine and Least Square Support Vector Machine Stock Forecasting Models in the Computer Science and Information Technology, Vol. 2, No.1, 2014, pp.30-39.

[13] Qisheng Yan, ShitongWang, and Bingqing Li, Forecasting Uranium Resource Price Prediction by Extreme Learning Machine with Empirical Mode Decomposition and Phase Space Reconstruction in the Hindawi Publishing Corporation, Discrete Dynamics in Nature and Society, 2014, pp.1-10.

[14] S.N. Sivanandam and S.N.Deepa, Principles of Soft Computing, Wiley India, 2007.